УДК 004.052.2

РАЗРАБОТКА ПЛАТФОРМЫ ТЕСТИРОВАНИЯ РАСПРЕДЕЛЁННЫХ СИСТЕМ

Коротков И.С.

Пермский государственный национальный исследовательский университет, г. Пермь, Российская Федерация

Научный руководитель: Постаногов И.С. – старший преподаватель кафедры МОВС

Аннотация. Разработана платформа тестирования распределенных систем, обладающая относительной простотой в использовании, низкой ценой эксплуатации и удобством использования при тестировании по принципу черного ящика. Разработанная платформа внедрена в учебный процесс и была использована для тестирования лабораторных работ студентов по курсу разработки распределенных систем.

Ключевые слова: распределенные системы, тестирование ПО, отказоустойчивость систем

Введение. Распределенные системы отличаются от традиционных систем тем, что их компоненты распределены по нескольким узлам сети и вынуждены общаться между собой по сети [1]. Сеть может быть нестабильной, имеет относительно большую задержку при передаче сообщений и ограниченную пропускную способность. Помимо этого, при сетевом взаимодействии на работу системы может повлиять изменение топологии сети, а также временная недоступность отдельных компонентов. Если не учитывать эти особенности еще во время проектирования распределенной системы, то она потребует большого количества исправлений и доработок и получится более сложной, чем могла бы [2].

Для того, чтобы убедиться в корректной работе распределенной системы в реальных условиях, её необходимо интенсивно тестировать. Причём тестирование требуется не только в промышленной разработке, но и при обучении студентов созданию распределенных систем в рамках образовательных курсов.

Инструмент, который мог бы помочь с тестированием в учебном процессе, должен быть дешёвым или бесплатным в использовании в некоммерческих целях, а также простым в настройке и применении при тестировании по принципу черного ящика (т. е. без настройки под каждую конкретную тестируемую систему). При этом необходимо, чтобы он поддерживал эмуляцию широкого спектра неисправностей, чтобы его применения было достаточно для всестороннего тестирования распределенной системы.

Рассмотренные нами решения либо обеспечивают широкий спектр функций, но являются сложными в использовании (*Jepsen, Fallout*), либо обеспечивают простоту настройки и эксплуатации, жертвуя при этом значительной частью функций (*Toxiproxy, Muxy*). Единственный инструмент, обладающий достаточным набором функций и низким порогом входа – *Gremlin* – является коммерческим и достаточно дорогим инструментом, который может позволить себе не каждая организация.

В связи с этим принято решение спроектировать и реализовать собственную платформу для тестирования, наиболее хорошо подходящую под наши требования.

Требования и технологии. При проектировании платформы к ней были предъявлены следующие требования:

– простота развертывания – разработанная платформа должна иметь настолько мало требований к окружению, насколько это возможно; она должна предоставлять наиболее простой способ развертывания, в котором возникновение случайной ошибки из-за особенностей окружения или ошибки пользователя будет сведено к минимуму;

– низкий порог входа при использовании – разработанная платформа должна обеспечивать наименьший возможный порог входа, позволяющий использовать ее без предварительного чтения документации или просмотра обучающих материалов;

– простота анализа результатов теста – разработанная платформа должна предоставлять результаты тестирования системы в едином месте в простом и понятном виде;

– поддержка широкого круга доступных для эмуляции неисправностей – сетевые неисправности (задержка, потеря и повреждение пакетов); неисправности, связанные с нехваткой вычислительных ресурсов (нехватка процессорного времени, ОЗУ, места на диске, истощение I/O);

– возможность тестирования распределенных систем по принципу черного ящика, т. е. без необходимости настройки платформы под каждую конкретную тестируемую систему и используемый ей набор технологий и протоколов коммуникации.

Помимо удовлетворения вышеприведенных требований, необходимо также написать руководство по использованию разработанной платформы, а также реализовать распределенную систему для использования её в качестве примера для тестирования в написанном руководстве.

В качестве платформы для запуска разработанной платформы используется *Kubernetes*, т. к. *Kubernetes* позволяет запускать распределенные системы с помощью простой и выразительной конфигурации, а также предоставляет возможность не привязываться к конкретным вычислительным узлам при проектировании и запуске распределенной системы, которая должна работать корректно даже при изменении топологии компьютерной сети (в том числе добавлении новых или удалении существующих вычислительных узлов). Кроме того, есть множество сервисов, позволяющих создать и использовать кластер *Kubernetes* в облаке, что позволяет пользователям работать с разработанной платформой даже при отсутствии у них достаточно производительного компьютера.

Проектирование и реализация. Разработана

Разработанная платформа состоит из 4 компонентов:

– *scheduler* (планировщик тестов) отвечает за генерацию тестовых сценариев для тестируемой системы, развернутой в том же кластере;

– *monitor* (монитор состояния теста) следит за тестируемой системой и завершает тест с ошибкой, если в работе тестируемой системы обнаружена ошибка;

– *workflows* (*API* для работы с запущенными тестами) предоставляет информацию об истории тестов и состоянии каждого теста для отображения в веб-приложении;

- *frontend* (веб-приложение) предоставляет графический пользовательский интерфейс.

Помимо этого, разработанная платформа использует *Litmus Chaos* для эмуляции неисправностей в компонентах тестируемой системы, а запуск теста осуществляется с помощью *Argo Workflows* – сервиса для выполнения последовательностей действий в кластере *Kubernetes.*

Все компоненты разворачиваются в кластере *Kubernetes* и взаимодействуют между собой согласно рисунку 1. Прерывистой линией показаны все внешние компоненты, непрерывной – компоненты разработанной платформы.

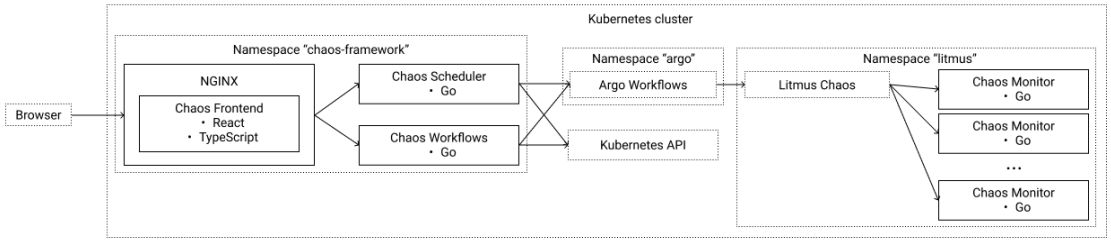

Рисунок 1 - Архитектура разработанной платформы

Платформа работает следующим образом:

1. пользователь открывает в браузере веб-приложение;

2. пользователь просматривает историю запущенных тестов: веб-приложение отправляет запрос в компонент *workflows*, который получает информацию о тестах из *Kubernetes API* и возвращает её;

3. пользователь создает новый тест: веб-приложение отправляет запрос в компонент *scheduler*, который генерирует по переданным параметрам описание теста для *Kubernetes* и запускает его;

3.1. компонент monitor запускается в ходе теста и следит за состоянием тестируемой распределенной системы;

4. пользователь просматривает состояние запущенного теста: веб-приложение отправляет запрос в компонент *workflows*, который возвращает состояние выбранного теста.

Внедрение. Перед внедрением разработанной платформы в учебный процесс необходимо написать для нее документацию и руководство по использованию, а также обозначить каналы коммуникации, по которым можно обратиться с вопросами, жалобами и предложениями по улучшению платформы.

Для документации и сбора сообщений об ошибках используется репозиторий на *GitHub*, а в качестве руководства служит документ в *Google Docs*. Руководство снабжено поясняющими иллюстрациями для каждого шага запуска разработанной платформы и тестирования распределенной системы.

Внедрение и практическое применение разработанной платформы проведено на студентах 4 курса механико-математического факультета ПГНИУ, имеющих в своём учебном плане курс «Технологии разработки распределенных систем», что позволило качественнее оценить полученные студентами в рамках изучения данной дисциплины навыки проектирования и разработки отказоустойчивых распределенных систем, а также развертывания и проведения тестирования на корректность работы и отказоустойчивость разработанных систем.

Несколько групп студентов изъявили желание воспользоваться разработанной платформой для тестирования разработанных ими систем, выявили и описали обнаруженные ошибки, а также исправили наиболее критичные из них. Протестированные системы были разработаны на разных языках и с применением различных технологий, но платформа благодаря своей универсальности работала для всех из них.

Заключение. Разработана, протестирована и внедрена платформа для тестирования распределенных систем, являющаяся бесплатной и относительно простой в использовании, а также позволяющая тестировать распределенные системы по принципу черного ящика.

Внедрение платформы показало её востребованность и эффективность при тестировании распределенных систем, разработанных студентами в рамках курса, посвященного этой теме.

Список литературы

1. *Burns, B. Distributed Systems / B. Burns — 1st ed. — O'Reilly Media, 2018. - 244 pp.*

2. *van Steen, M. Distributed Systems / M. van Steen, A. S. Tanenbaum — 3rd ed. — CreateSpace Independent Publishing Platform, 2018. - 596 pp.*

UDC 004.052.2

DISTRIBUTED SYSTEMS TESTING PLATFORM

Korotkov I.S.

Perm State National Research University, Perm, Russian Federation

Postanogov I.S. – senior lecturer of the department of MSCS

Annotation. We developed a platform for the distributed systems testing which is easy and cheap to use, and is suitable for black box testing. The platform was integrated in the educational process and was used for testing students' homework in the course of distributed systems development.

Keywords: distributed systems, software testing, system reliability